A Spectrum of IV&V Modeling Techniques

Collection of Case Study Material[1]

1. Modeling Language and Models Collected by the University of Minnesota (UMN)

A collection of five models, representing increasingly complex flight guidance systems (FGS), has been obtained from Rockwell Collins, Inc. These models are written in UMN's RSML^{-e} notation, for which UMN tools are available to produce output suitable for the SMV and PVS verification tools. UMN has determined that the models are good candidates for case studies in this project; they are 1) relevant to NASA's mission, 2) they are real, or at least realistic, and 3) they are complex enough to stress the capabilities we propose to investigate. Also, because tools have already been written to integrate RSML^{-e} and popular software verification tools, we will be able to compare the new and unproven WVU / NASA modeling language and tools' effectiveness to the results of established techniques.

For detailed information about the UMN models, see the report "Collect Models from UMN Clients" (umn_models.pdf). For more information about RSML^{-e}, see "Definition of UMN Languages: RSML^{-e}" (umn_languages.pdf) and "Definition of UMN Test Engines" (umn_test_engines.pdf).

2. Modeling Language and Models Collected by West Virginia University (WVU), NASA IV&V

Models are written in a simple input language based on the "communicating finite-state machines" (FSM) formalism underlying many specification languages and model checker input languages. The specific format is described in "Definition of WVU Languages" (wvu_languages.pdf). For an overview of this FSM language, the search technique associated with it in past research, and brief results motivating the current work, see "An Alternative to Model Checking: Verification by Random Search of AND-OR Graphs Representing Finite-State Models" (alternative.pdf).

We will investigate Livingstone, a software system designed to support unmanned spacecraft's automatic reconfiguration and error recovery. Detailed information about Livingstone is available online at "http://ic.arc.nasa.gov/projects/mba/projects-

[1] Because this document was delivered behind schedule, it contains information up-to-date 10/19/02, and is identical to the documents "Assessment of Potential Case Studies" and "Selection of Case Study" delivered at the same time.

/livingstone.html," an excerpt from which follows:

Livingstone accepts a model of the components of a complex system such as a spacecraft or chemical plant and infers from them the overall behavior of the system. Livingstone also notes which commands are being given to the system and what observations are available. From this, Livingstone is able to monitor the operation of the system, diagnose its current state, determine if sensors are giving impossible readings, recommend actions to put the system into a desired state even in the face of failures and so on.

Because Livingstone reasons about explicit models of the system it is interacting with, rather than following a program or rules, a Livingstone-based controller is highly capable, flexible and easy to maintain. Livingstone also takes into account all available information and observations, drawing conclusions which reach across a complex system in a way which would be difficult for a traditional software system or time consuming for a human operator.

Livingstone is able to perform significant deduction in the sense / response loop by drawing on our past experience at building fast propositional conflict-based algorithms for model-based diagnosis, and by framing a model-based configuration manager as a propositional feedback controller that generates focused, optimal responses. Livingstone's representation formalism achieves broad coverage of hybrid hardware / software systems by coupling the transition system models underlying concurrent reactive languages with the qualitative representations developed in model-based reasoning. Livingstone automates a wide variety of tasks using a single model and a single core algorithm, thus making significant progress towards achieving a central goal of model-based reasoning.

See also "A Model-Base Approach to Reactive, Self-Configuring Systems" (livingstone.pdf).